NetArrays Project Program Development Example

NetArrays Project Program Development Example

RTP Corporation
2832 Center Port Circle
Pompano Beach, FL 33064
Phone: (954) 597-5333
Internet: http://www.rtpcorp.com

File Name: NetArrays Example 3000.pdf
Last Updated: 2/26/18

NetArrays Project Program Development

The RTP3000 system supports both safety and control in a single application. The example provided in this document demonstrates how easy it is to use this configuration tool. It steps you though the process of creating SIF's, adding control and providing a secure location for HMI/Operator access.

In a matter of minutes you are able to create hundreds of points for programming. We will configure a traditional QMR system for safety and add a separate chassis for critical I/O to do control. The safety system will consist of 64 Triple redundant analog inputs and 16 dual redundant digital outputs. The DCS will consist of 32 digital inputs and 12 digital outputs.

The only equipment required is a PC or laptop with NetSuite installed. To verify the project program we will run on the NetArrays built-in Simulator.

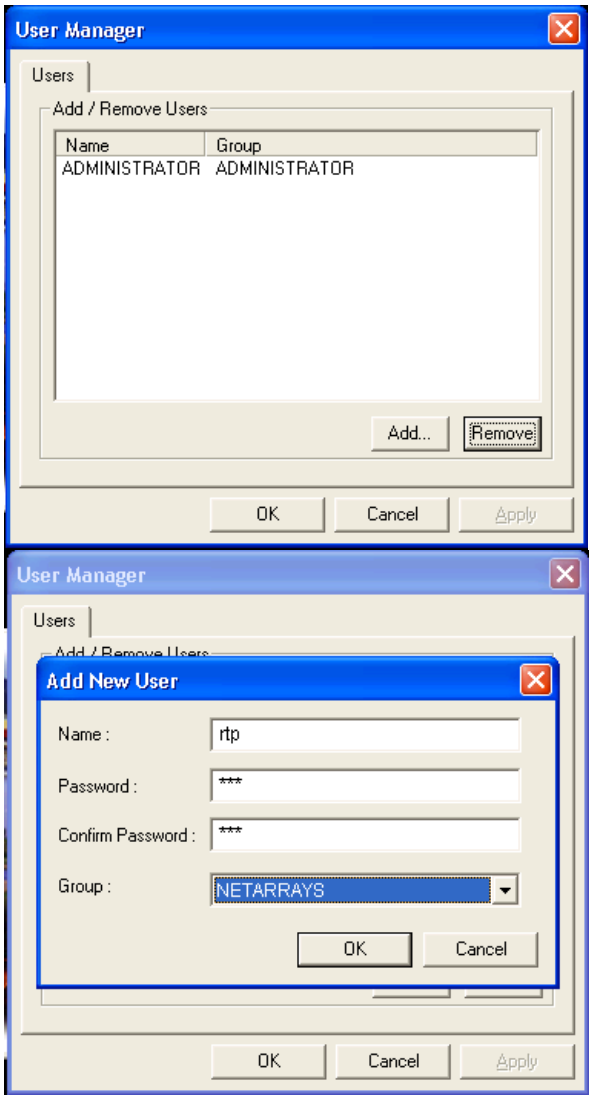
Note that the NetArrays program developed here will be used by the Setup Examples "RTPView Project Program" and "Redundant Alarm Monitoring Configuration."

Start NetArrays

Start NetArrays

- On your PC, press **Start** and select **Programs ▸ RTP NetSuite ▸ NetArrays**.
- When executing the very first time, create a username/password as follows:
- Type "administrator" for User Name and "admin" for Password as shown below and press the **User Manager** button.

- Press the **Add**… button at the bottom to create a NetArrays user account:



- For this example, a generic name of “rtp” is used above. Password is “rtp”. You may choose your own name and password. The group selected is NETARRAYS.
- Press OK button on the subsequent two screens.
- Log in using the new User Name “rtp” and Password “rtp” as shown below and press the OK button.

- Click **Open Project**.
- Select **My_First**", and Click **OK**.

I/O Configuration

Add I/O Cards to the I/O Configuration Form

- In the I/O Configuration form, expand the **Rack 00=Rack9D** Rack by clicking on the ⊞. Open the box of RTP **Analog Cards** in the I/O Configuration Toolbox by clicking on the ⊞.

Create the SIF Portion of the IOC

- Add the following for the SIF portion of the project. The DCS portion will be added later.

Add 32 Channel AI Card

- Drag and drop a 3126 32-Ch Scanning Single-ended AI icon from the I/O Configuration Toolbox RTP Analog Cards into slot **6** of the QMR chassis.
- In the Auto Tag Generation window, enter the **Use Card Specific Prefix** of **AI1**_. Then click the **OK** button.

  Note- Redundant Inputs are voted using the signal validation table. Please refer to signal validation help for more information.

Add another redundant 3126 32-Channel AI Cards to make a Dual Redundant Configuration

- Position the cursor over the AI card added to the I/O Configuration Form and click the right mouse button.
- Select **Copy As Redundant** from the menu.
- Move the cursor to click on the **Slot 07=Empty** position. Right click and select **Paste.**

First Analog Input Card Property

Property Manager - 3126 - 32 Channel AI()

Card Properties

Card	Slot 06=3126 - 32 Channel AI
Float Cal High 00	(AI_CH0A)
(Tag)	AI_CH0A
Float Cal High 01	(AI_CH1A)
(Tag)	AI_CH1A
Float Cal Low 00	(AI_CL0A)
(Tag)	AI_CL0A
Float Cal Low 01	(AI_CL1A)
(Tag)	AI_CL1A
Float Board Temp	(AI_TMPA)
(Tag)	AI_TMPA
Integer Card Revision	(AI_CRA)
(Tag)	AI_CRA
Integer Error Detection	(AI_EDA)
(Tag)	AI_EDA
Integer Card Status 00	(AI_CS0A)
(Tag)	AI_CS0A
Integer Card Status 01	(AI_CS1A)
(Tag)	AI_CS1A
Integer Channel Error Status 00	(AI_CE0A)
(Tag)	AI_CE0A
Integer Channel Error Status 01	(AI_CE1A)
(Tag)	AI_CE1A

I/O Channel Properties

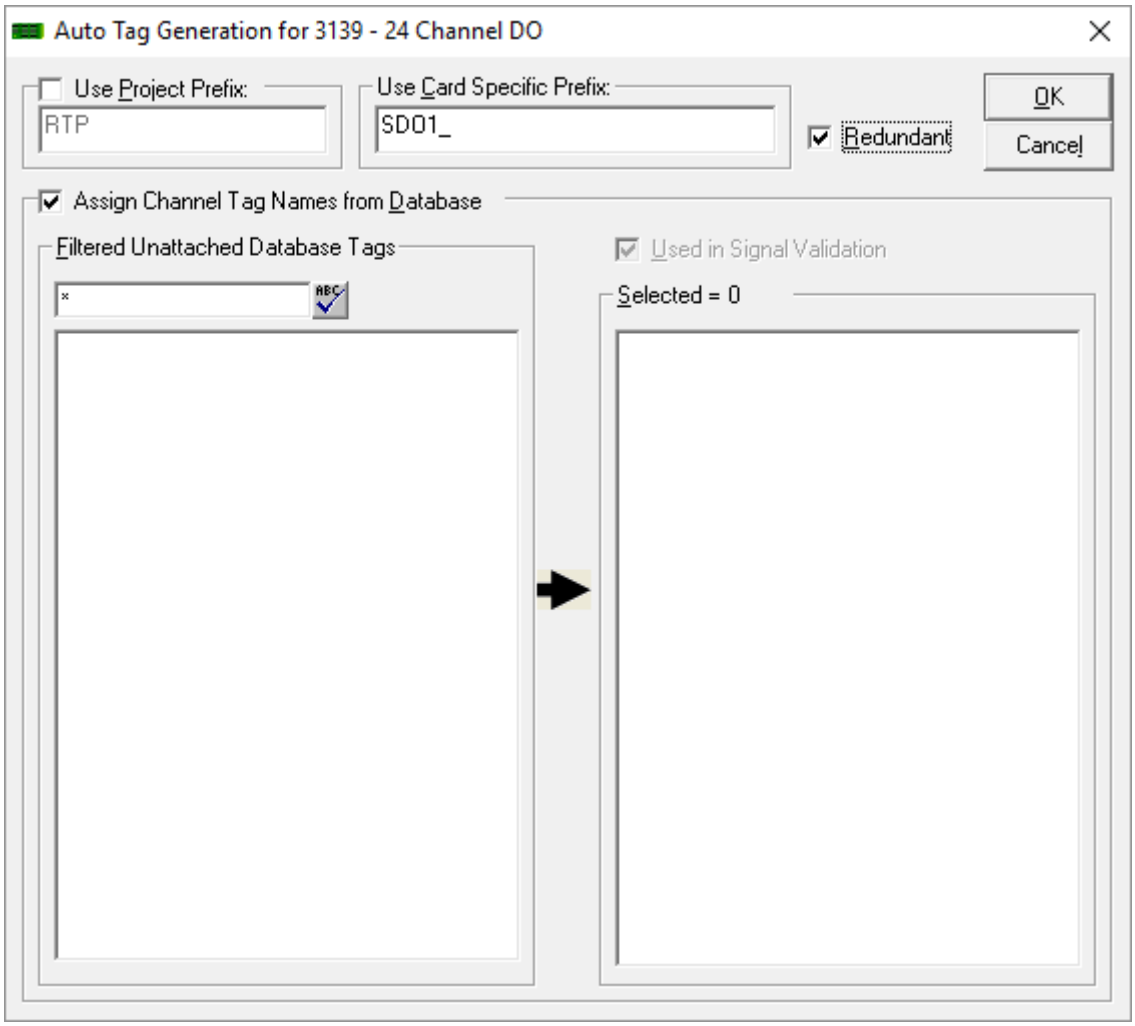
	Channel	I/O Tag	Filter	Threshold	SOE Low	SOE High	Guard Band Low	Guard Band High
Float	Input 00	AI_I00A	10	10	0	0	-10.1	10.1
Float	Input 01	AI_I01A	10	10	0	0	-10.1	10.1
Float	Input 02	AI_I02A	10	10	0	0	-10.1	10.1
Float	Input 03	AI_I03A	10	10	0	0	-10.1	10.1
Float	Input 04	AI_I04A	10	10	0	0	-10.1	10.1
Float	Input 05	AI_I05A	10	10	0	0	-10.1	10.1
Float	Input 06	AI_I06A	10	10	0	0	-10.1	10.1
Float	Input 07	AI_I07A	10	10	0	0	-10.1	10.1
Float	Input 08	AI_I08A	10	10	0	0	-10.1	10.1
Float	Input 09	AI_I09A	10	10	0	0	-10.1	10.1
Float	Input 10	AI_I10A	10	10	0	0	-10.1	10.1
Float	Input 11	AI_I11A	10	10	0	0	-10.1	10.1
Float	Input 12	AI_I12A	10	10	0	0	-10.1	10.1
Float	Input 13	AI_I13A	10	10	0	0	-10.1	10.1
Float	Input 14	AI_I14A	10	10	0	0	-10.1	10.1
Float	Input 15	AI_I15A	10	10	0	0	-10.1	10.1
Float	Input 16	AI_I16A	10	10	0	0	-10.1	10.1
Float	Input 17	AI_I17A	10	10	0	0	-10.1	10.1
Float	Input 18	AI_I18A	10	10	0	0	-10.1	10.1
Float	Input 19	AI_I19A	10	10	0	0	-10.1	10.1
Float	Input 20	AI_I20A	10	10	0	0	-10.1	10.1
Float	Input 21	AI_I21A	10	10	0	0	-10.1	10.1

- Right-click on the Analog Input card's icon in Slot 06 to display the menu and select **Properties** to open the card's Property Manager window.

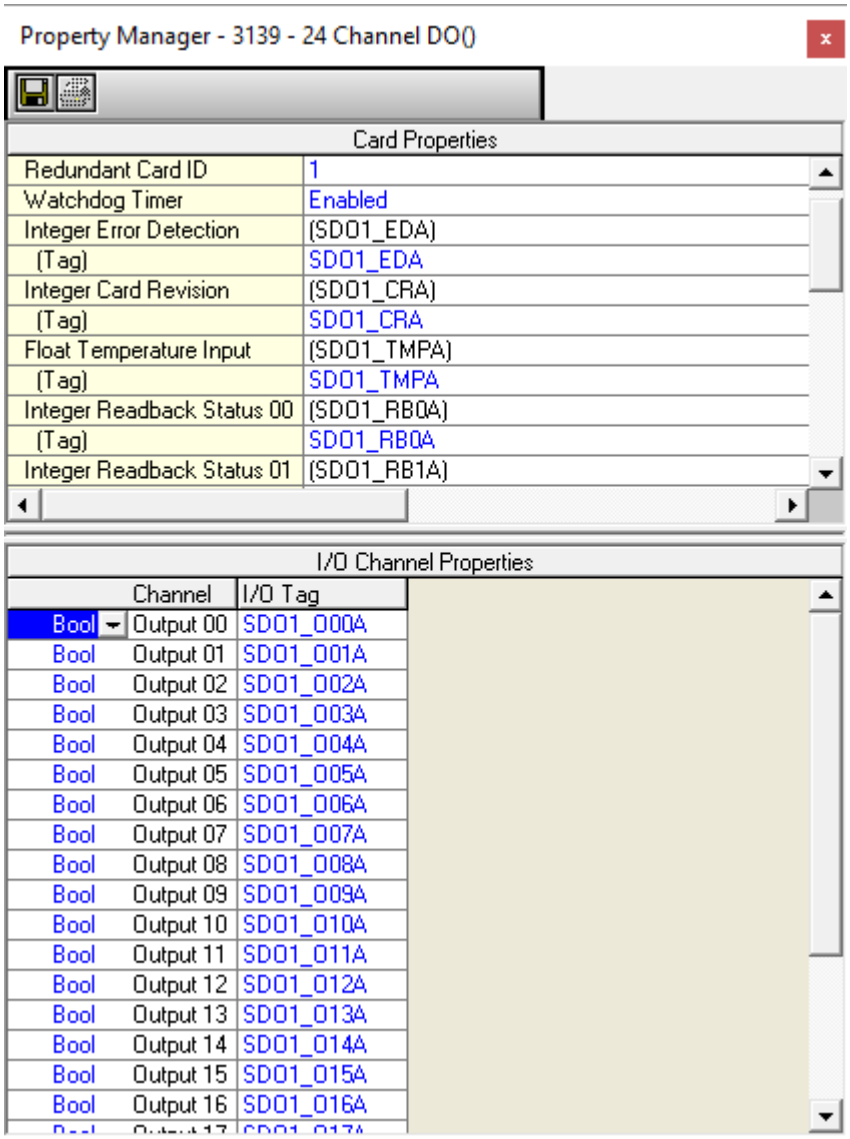
- Observe that the Postfix **A** has been added for each input parameter's tagname. The I/O card properties including diagnostics are listed at the top and the I/O channel information is listed below.
- Close the Property Manager window by clicking on .
- Redundant Analog Input card in slots 07 has the Postfix **B** for the I/O Card Property and I/O Channel Property tagnames.

Add Two Digital Output Cards

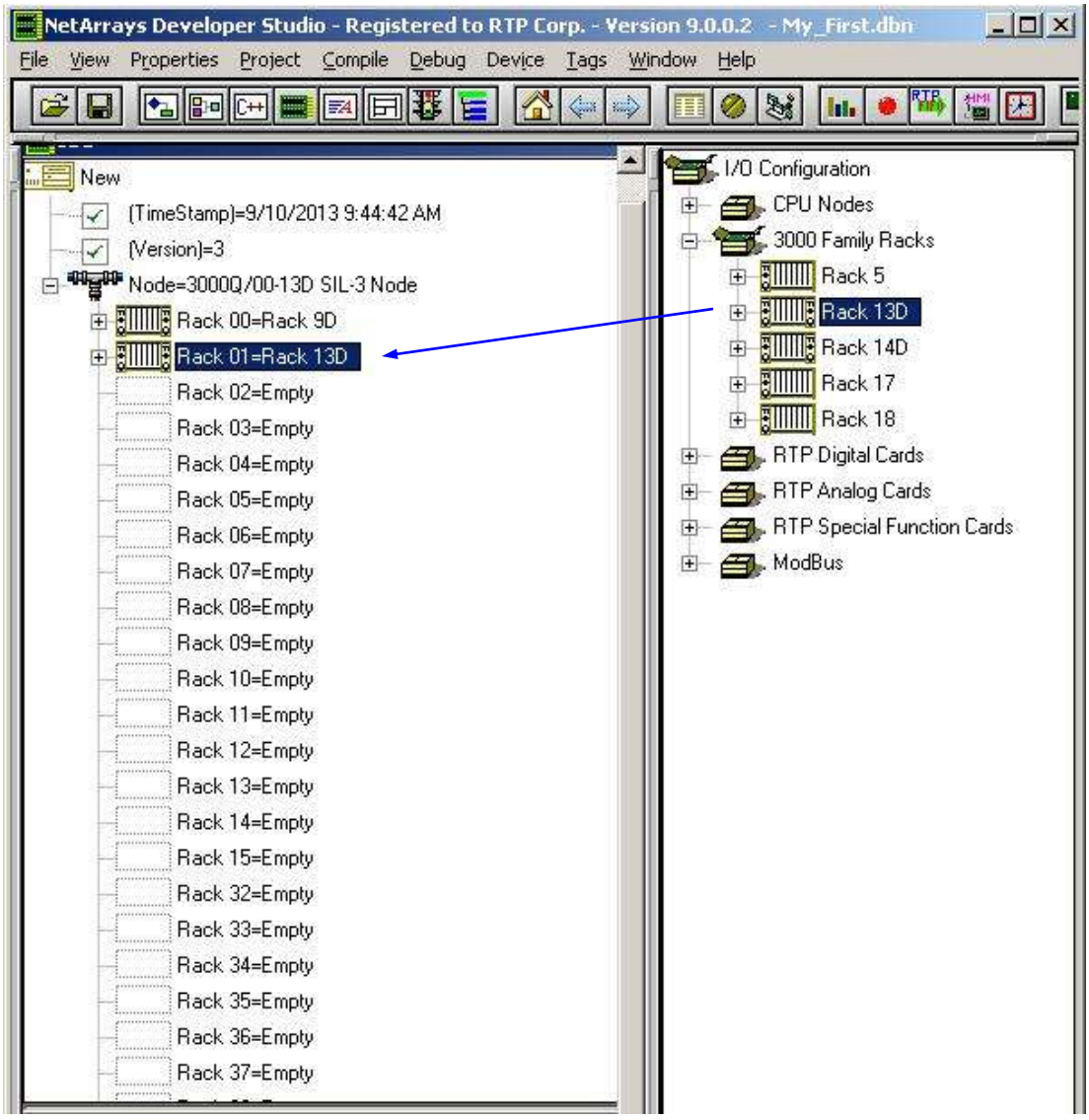
- Open the box of RTP **Digital Cards** in the I/O Configuration Toolbox by clicking on the ⊞.
- Drag and drop the **3139 – 24 Channel DO** card from the I/O Configuration Toolbox RTP Digital Cards into slot 09.
- In the Auto Tag Generation window, enter the Prefix **SDO1**_ and select **Redundant**. Click **OK**.



- Right click on the card added and select Properties. Note that the Redundant Card ID is set to **1**. Each time a set of redundant cards is added, a unique Card ID will be supplied.



- Copy the card as redundant and paste into slot **10**.



- Open the **3000 Family Racks** box in the I/O Configuration Toolbox by clicking on the ⊞.
- Drag and drop a **Rack 13D** chassis from the **3000 Family Racks** into the **Rack 01=Empty** location.
- Open Rack **01** by clicking on the ⊞.

Add a Digital Input Card

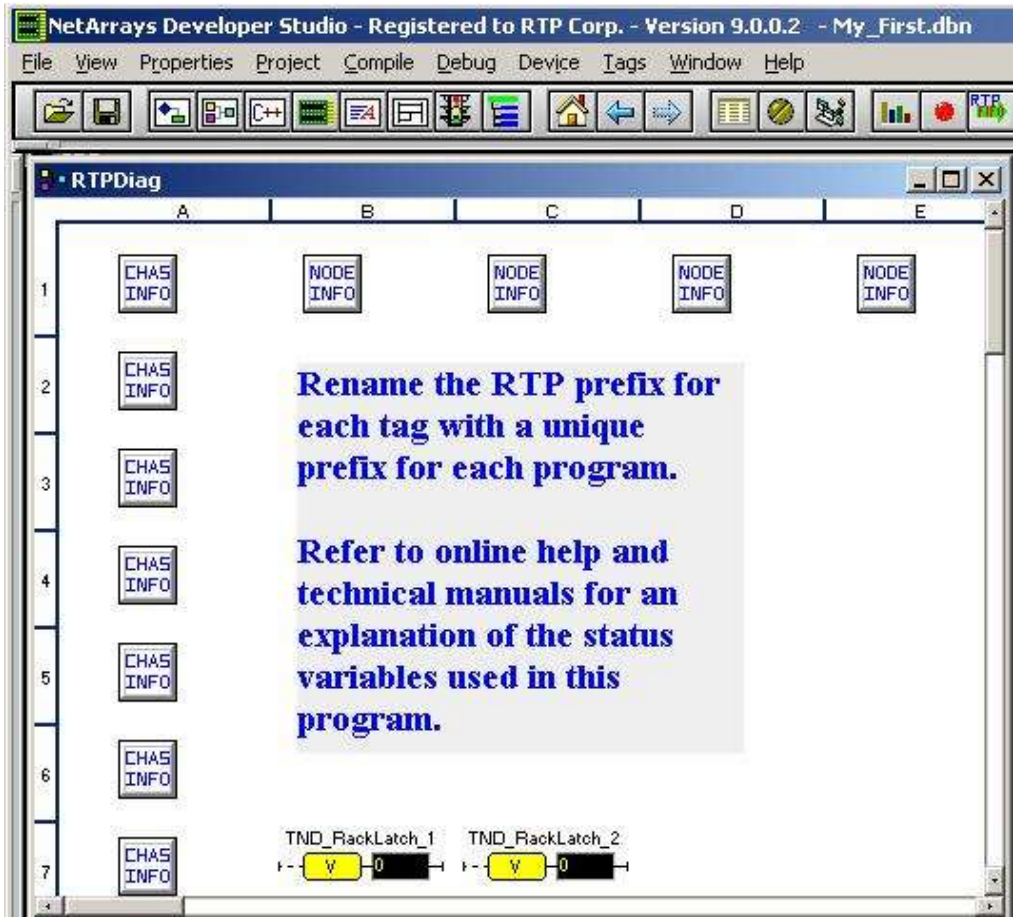
- Open the box of RTP Digital Cards in the I/O Configuration Toolbox by clicking on the ⊞.
- Drag and drop the **3115 - 32-Channel DI** card into slot **00**.
- In the Auto Tag Generation window, enter the Prefix **DI1**_. Click **OK**.

Add an Analog Output Card

- Open the box of RTP Analog Cards in the I/O Configuration Toolbox by clicking on the ⊞.
- Drag and drop the **3121 16-Channel Analog Output** card into slot **01**.
- In the Auto Tag Generation window, enter the Prefix **AO1**_. Click **OK**.

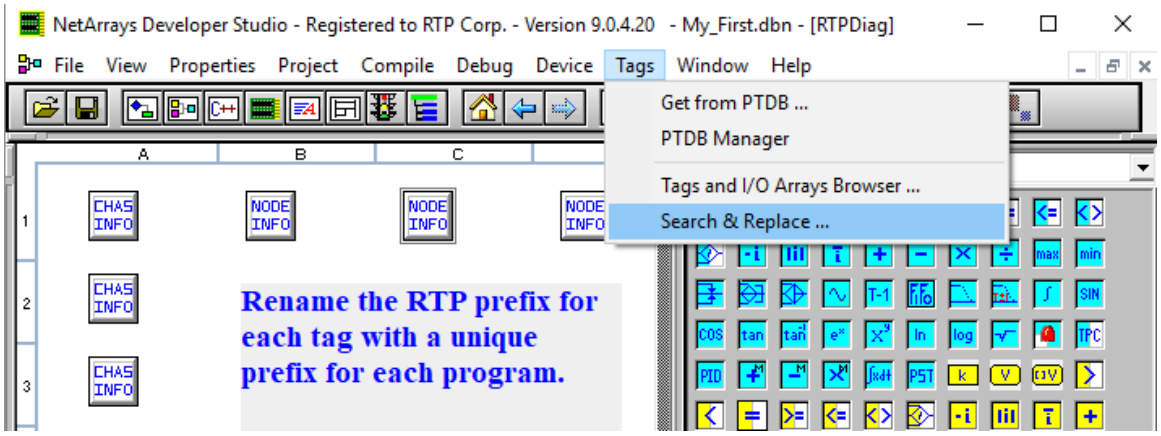
RTP Diagnostics

- Display the **RTP_Diag** Form by clicking on the button in the Main Toolbar, then double clicking on the RTP_Diag form object..

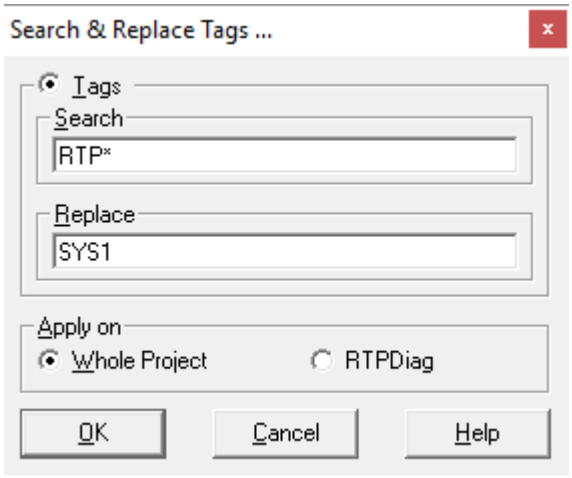


Diagnostics variables are pre-configured for each node and each chassis processor. Simply select and delete any Node Info object or Chassis Info object that are not needed. This can be accomplished by using the mouse. Left click and draw a box around the un-used Node Info or Chassis Info object and then press the delete key.

- Double click the **CHAS INFO** object to observe chassis diagnostics.
- Double click the **NODE INFO** object to observe node processor diagnostics.
- Delete objects from cell **A3** to **A16** as they are un-used. This will leave 4 **NODE INFO** objects for Node Processors **A**, **B**, **C**, and **D** as well as 2 **CHAS INFO** objects for the 2 I/O Chassis.
- Using the tags search and replace feature customizes diagnostics variables. On the menu bar, click on Tags, search and replace.

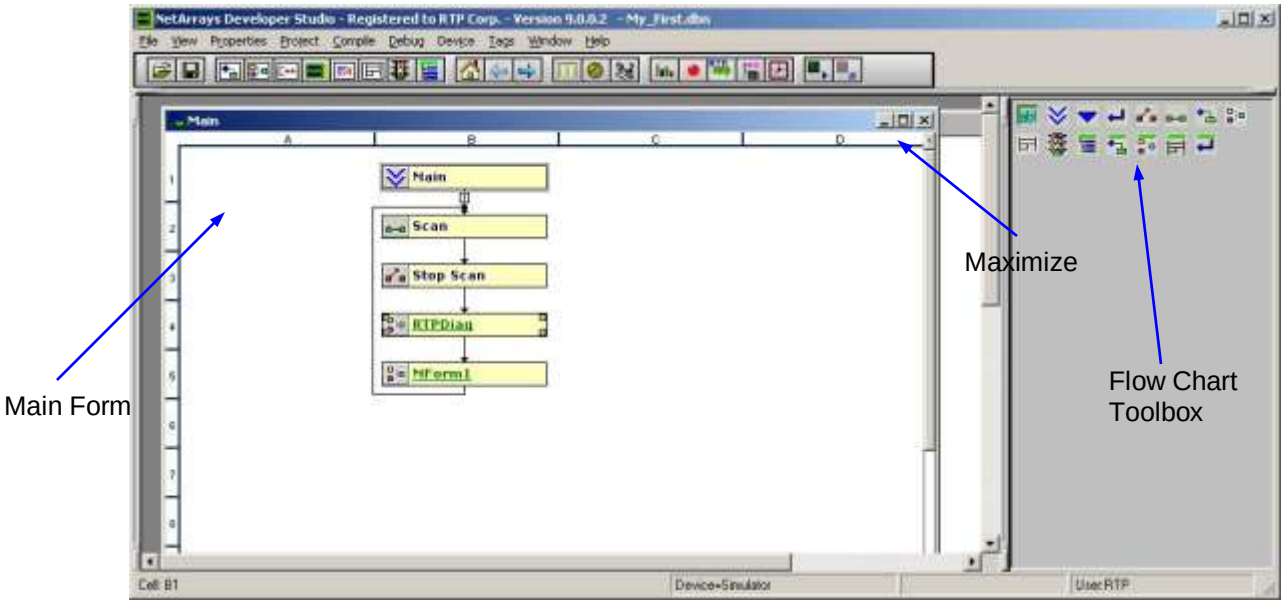


- In **Search & Replace Tags…** window, enter **RTP*** in the **Search** field and **SYS1** in the **Replace** field.



- Observe approximately 139 variables have been renamed.
- Select the Tags menu again and click on Tags and I/O Arrays Browser.
- Select the tag column and sort Z-A. Observe you have the SYS1 diagnostic
- Observe 399 total tagnames have been generated.

NetArrays Main Form



- Click on the Maximize button ☐ in the the banner of the Main Form. If the Maximize button is not visible on the Main Form, double click on the Main banner. The Main Form is a 16x16 grid. Columns A – P and Rows of 1 – 16.
- The module forms that are part of a SIF may only contain SIL I/O points. All the variables inside the SIF forms will be marked as “read only” (R/Only True). Therefore there is protection provided against any HMI or operator having access to alter or interfere with the safety integrity function. The HMI is **NOT** provided write access to “read only’ variables. The HMI can only access Read/Write (R/Only False) variables.
- Where necessary, the engineer programmatically may provide writable access for the HMI to selected variables, outside of the SIF forms. These forms have form properties with PartOfSIF and VariableROnly set to False.

Order of Execution

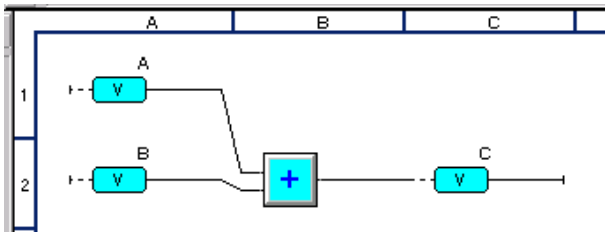
Note that every Module Form has an ExecutionMode property that defaults to “Horizontal” but can be changed to “Vertical.” This specifies the order in which the Objects are executed on the Form.

The first cell evaluated is always cell A1.

When ExecutionMode is set to “Horizontal” the cells are evaluated in the following order: A1, B1, C1,…,A2, B2, C2,…,A16, B16, C16,…,N16, O16, P16.

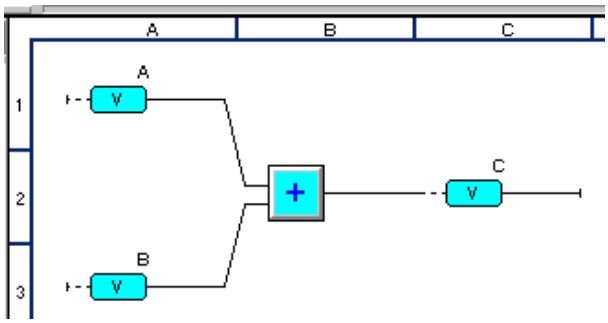
When ExecutionMode is set to “Veertical” the cells are evaluated in the following order: A1, A2, A3,…,B1, B2, B3,…,P1, P2, P3,…,P14, P15, P16.

For Example



Example 1

For Horizontal Execution Mode, Example 1 may not get the same result as Example 2, as the value of “B” used below will be one scan later.

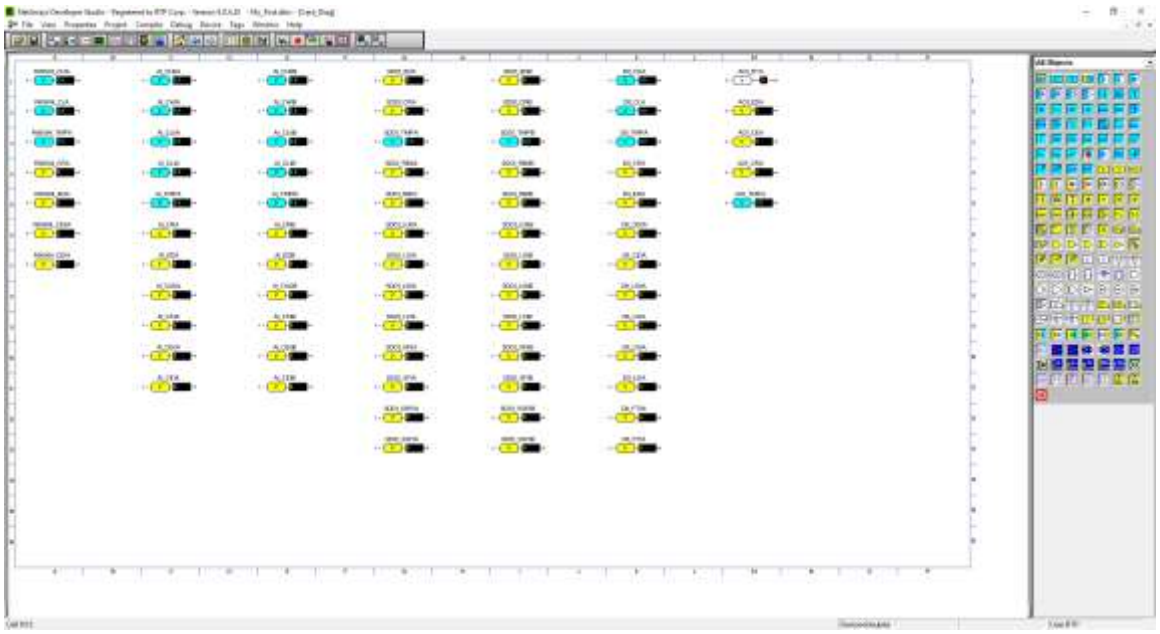


Example 2

Both of these examples, however, will have the same result for Vertical execution mode.

Developing Diagnostic Page

- Card property variables can be broken down to individual bits to specify the error/channel in error on a card.
- Drag a Module Object icon from the Toolbox and onto cell B6, Note that NetArrays automatically assigns the Tag MForm2.
- Change Param/Tag to Card_Diag.
- Double click Card_Diag and click yes to create the page.
- Individually, copy each I/O card as is from the IOC page and paste into the Card_Diag form.
- Once complete, the page should look as below.



NetArrays Logic Development

- Return to the Main Form by clicking on the button in the Main Toolbar.
- Drag a Module Object icon from the Toolbox and onto cells B7, B8, and B9. Note that NetArrays automatically assigns them Tags MForm2, MForm3, and MForm4.

The Main Form must be constructed as one continuous program loop. To include the new Module Form object into the loop, you must rearrange the connections between the objects.

- Move the curser to the output of MForm1. You will know that the cursor is in the right spot when the small square appears on the output connector. Then click on the small square to disconnect the connection between the MForm1 object and the Scan object.
- Move the curser to the output of MForm1 again. Left mouse click, hold the mouse button down and drag the cursor to the input of the Card_Diag object. Release the mouse button to complete the connection. Repeat the operation until all Forms are interconnected. The Main Form should now look like the following figure.

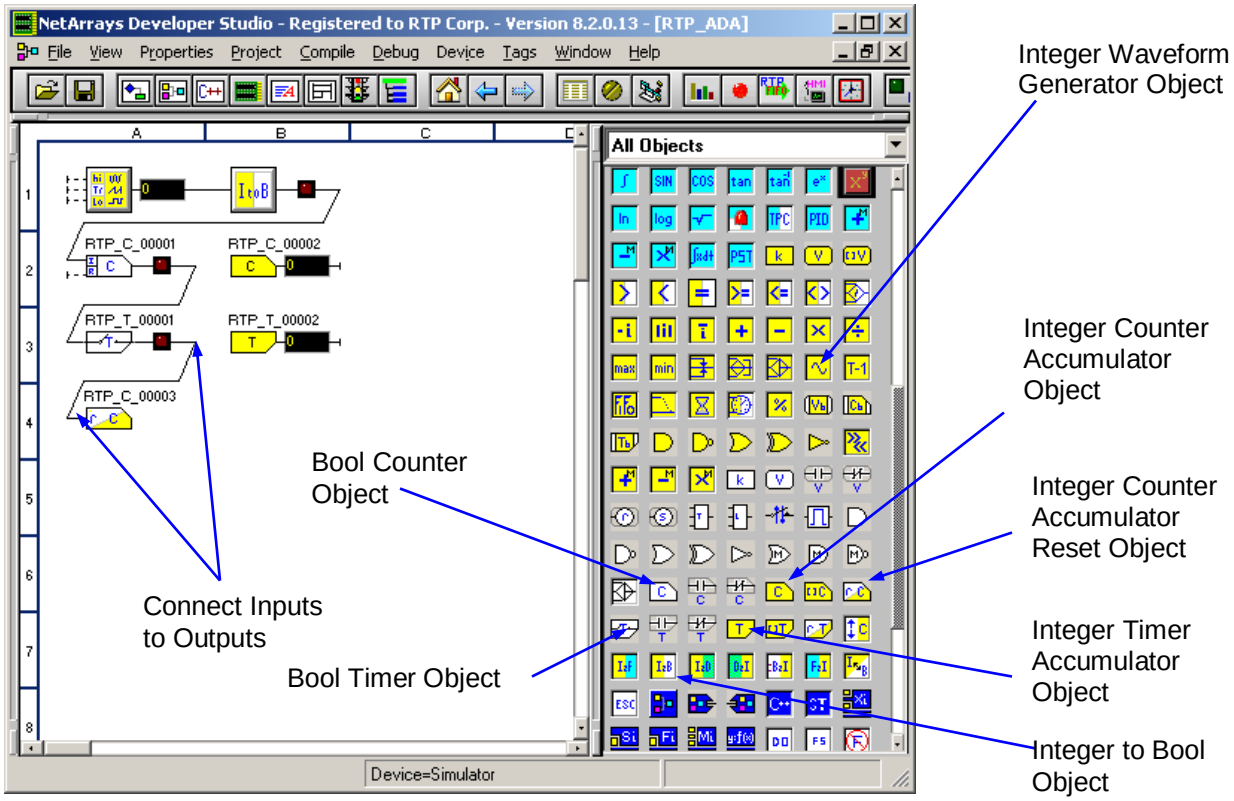
  Note: The **Scan** and **Stop Scan** objects are required in the Main Form, and they must remain within the program's main loop as indicated.
- Right-click on the **MForm2** object to display the menu and select **Properties** to open the object's Property Manager window.
- Click on the **(Tag)** field and type in the name Counter followed by a Return. The Tag **Counter** will appear on the object in the Main Form and in the Project Explorer. Close the Property Manager window.
- Changes **MForm3** to RTP_View and **MForm4** to RTP_ADA.

The next step is to define the logic within the Module Forms. Module Forms open as a 16x16 grid. Columns A – P and Rows 1 – 16.

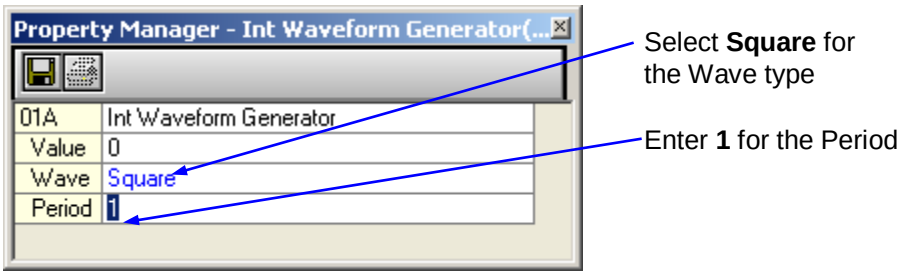
Module Form "Counter"

Add Objects to a New Module Form to Demonstrate Counters and Timers
You will now create a new Module Form and place objects onto it from the Toolbox. This Module Form illustrates how NetArrays counters and timers function.

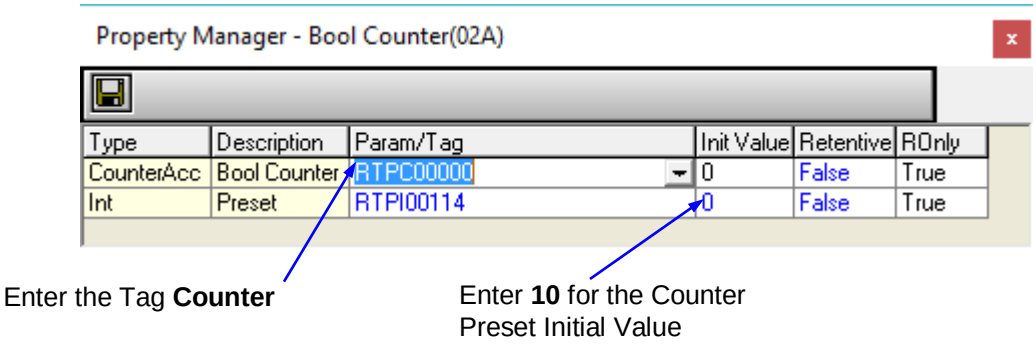
- Return to the Main Form by clicking on the button in the Main Toolbar.
- Double-click on the **Counter** object in the Main Form. The following prompt will appear on your screen.
- Click on the **Yes** button to create a new Module Form with the Tag **Counter**.
- Drag an Integer Waveform Generator object icon from the Toolbox and drop it onto cell A1. This object will be the source of the pulses we will be counting.
- Drag a Bool Integer to Bool object icon from the Toolbox and drop it onto cell B1. This object converts integer values into Boolean values.
- Drag a Bool Counter object icon from the Toolbox and drop it onto cell A2. This object counts a preset number of input pulses.
- Drag an Integer Counter Accumulator object icon from the Toolbox and drop it onto cell B2. This object stores the number of pulses counted.
- Drag a Bool Timer object icon from the Toolbox and drop it onto cell A3. This object times a preset time period.
- Drag an Integer Timer Accumulator object icon from the Toolbox and drop it onto cell B3. This object stores the time period counted.
- Drag an Integer Counter Accumulator Reset object icon from the Toolbox and drop it onto cell A4. This object resets the Counter.



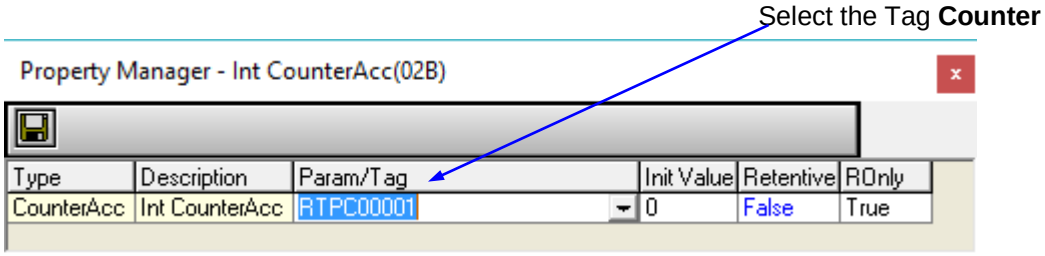
- Connect the input of the Integer Counter Accumulator Reset object to the output of the Bool Timer object. Connect the input of the Bool Timer object to the output of the Bool Counter object. Connect the input of the Bool Counter object to the output of the Integer To Bool object. Connect the input of the Integer To Bool object to the output of the Integer Waveform Generator.
- Double-click on the Integer Waveform Generator object. Click on the **Wave** field in the Property Manager and select **Square** from the list. Click on the **Period** field and enter **1** followed by a Return. Close the Property Manager window.



- Double-click on the Bool Counter object. Click on the **(Tag)** field in the Property Manager and enter the name **Counter**, followed by a Return. Click on the Counter Preset **Initial Value** and enter the value **10** followed by a Return. (The Counter will count 10 pulses before it switches its output to True.) Close the Property Manager window.



- Double-click on the Integer Counter Accumulator object. Click on the **(Tag)** field in the Property Manager and select **Counter** from the list of Tags. (The counter object and its accumulator must have the same Tag.) Close the Property Manager window.

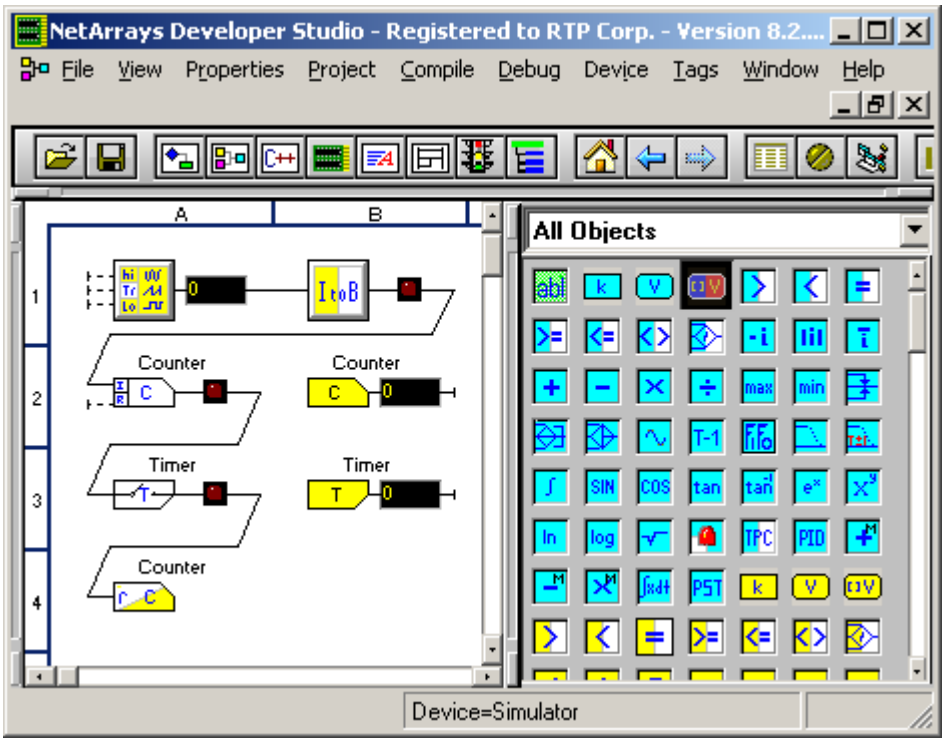


- Double-click on the Bool Timer object. Click on the **(Tag)** field in the Property Manager and enter the name **Timer**, followed by a Return. Click on the Timer Preset **Initial Value** and enter the value **20** followed by a Return. Make sure that

the **Period** field indicates **Second**. (The Timer will wait 20 seconds before it switches its output to True.) Close the Property Manager window.

- Double-click on the Integer Timer Accumulator object. Click on the (Tag) field in the Property Manager and select Timer from the list of Tags. (The timer object and its accumulator must have the same Tag.) Close the Property Manager window.
- Double-click on the Integer Counter Accumulator Reset object. Click on the **(Tag)** field in the Property Manager and select **Counter** from the list of Tags. (The reset object must have the same Tag as the counter it is resetting.) Close the Property Manager window.

The Module Form should now look like the following example.



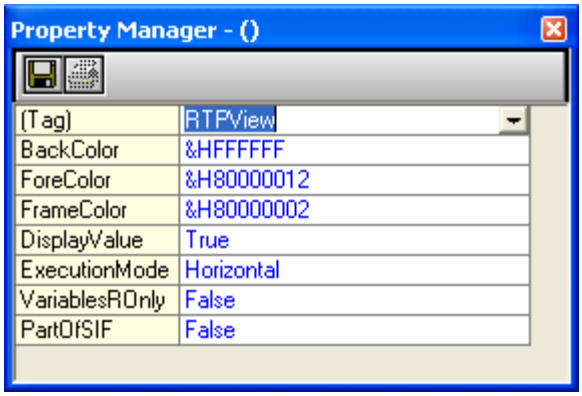
- In the NetArrays **File** menu, select **Save My_first.dbn**. The project file was originally created when the New Project was created.

This completes the Counter Module Form. The next step is to define the logic within the RTP_View Module Form.

Module Form "RTP_View"

Add Objects to a New Module Form to provide signals for RTP_View Example
You will now create a new Module Form and place objects onto it from the Toolbox.

- Return to the Main Form by clicking on the button in the Main Toolbar.
- Double-click on the **RTP_View** object in the Main Form. The following prompt will appear on your screen.
- Click on the **Yes** button to create a new Module Form with the Tag **RTP_View**.
- Right click on the page and select **Properties** to open the Project Property window.
- In the Property Manager window click on **PartOFSIF**, click and select **False**. Click on **VariablesROnly**, click and select **False**.



- Drag an Integer Constant object to Cell A1. Enter the Tag name **Three** and an Initial Value of **3**. Close the Property Manager window. This object will set the upper limit of a Waveform Generator.
- Drag another Integer Constant object to Cell A2. Enter the Tag name **Zero**. Close the Property Manager window. This object will set the lower limit of a Waveform Generator.
- Drag a Bool Variable object to Cell A3. Enter the Tag name **Key**. Close the Property Manager window. The output of this object will be controlled by a Key Switch object in the RTPView project.
- Drag an Integer Waveform Generator object to Cell B2. Select **Ramp Up** for the **Wave** function. Enter **0.5** for the **Period**. Close the Property Manager window. Connect the High Limit input of the Waveform Generator to the output of Integer Constant **Three**. Connect the Low Limit input of the Waveform Generator to the output of Integer Constant **Zero**. The output of the Waveform Generator will run the Mixer animation in the RTPView project.
- Drag an Integer Gate object to Cell C3. Connect the Input 1 input of the Gate object to the output of the Waveform Generator. Connect the Control input of the Integer Gate object to the output of Bool Variable Key. Connect the Input 2 input of the Integer Gate object to the output of Integer Constant Zero. This object activates the Mixer object when the Key Switch is turned on.
- Drag an Integer Variable object to Cell D3. Enter the Tag name **Mixer**. Close the Property Manager window. Connect the object's input to the output of the Integer Gate. This object will pass the output of the Waveform Generator, or zero, to operate the Mixer object in the RTPView project.

- Drag a Float Variable object to Cell A5. Enter the Tag name **Fader**. Close the Property Manager window. The Fader object in the RTPView project will control this variable.
- Drag a Float Constant object to Cell A6. Enter the Tag name **OneHundred** and an Initial Value of **100**. Close the Property Manager window. This object will set the high limit of a Float Ratio object.
- Drag a Float Constant object to Cell A7. Enter the Tag name **FloatZero** and an Initial Value of **0.0**. Close the Property Manager window. This object will set the low limit of a Float Ratio object.
- Drag a Float Lead/Lag object  to Cell B5. Under the Gain heading, enter an Initial Value of **1**. Under the Lag heading, enter an Initial Value of **0.2**. Leave the Lead Initial Value at the default value (**0**). Close the Property Manager window. Connect the Input of the Lead/Lag Filter to the output of the Float Variable **Fader**. This object drives the fill level of a Tank in the RTPView project.
- Drag a Float Ratio object  to Cell B7. Under the Output Max heading, enter an Initial Value of **100**. Under the Output Min heading, enter an Initial Value of **-40**. Close the Property Manager window. Connect the High Limit input of the Ratio object to the output of the Float Constant **OneHundred**. Connect the Input of the Ratio object to the output of the Float Variable **Fader**. Connect the Low Limit input of the Ratio object to the output of the Float Constant **FloatZero**. The Ratio object converts the range of the Fader object input (0 to 100) to a temperature range of –40 to 100 for the Temperature Gauge object in the RTPView project.
- Drag a Float Variable object to Cell C5. Enter the Tag name **TankLevel**. Close the Property Manager window. Connect the input to the output of the Lead/Lag Filter object. This variable will control the level displayed in the Tank object in the RTPView project.
- Drag a Float Variable object to Cell C7. Enter the Tag name **Celsius**. Close the Property Manager window. Connect the input to the output of the Float Ratio object. This variable will simulate a temperature input to the Temperature Gauge in the RTPView project.
- In the NetArrays **File** menu, select **Save My_first.dbn** to save your project.

Module Form “RTP_ADA”

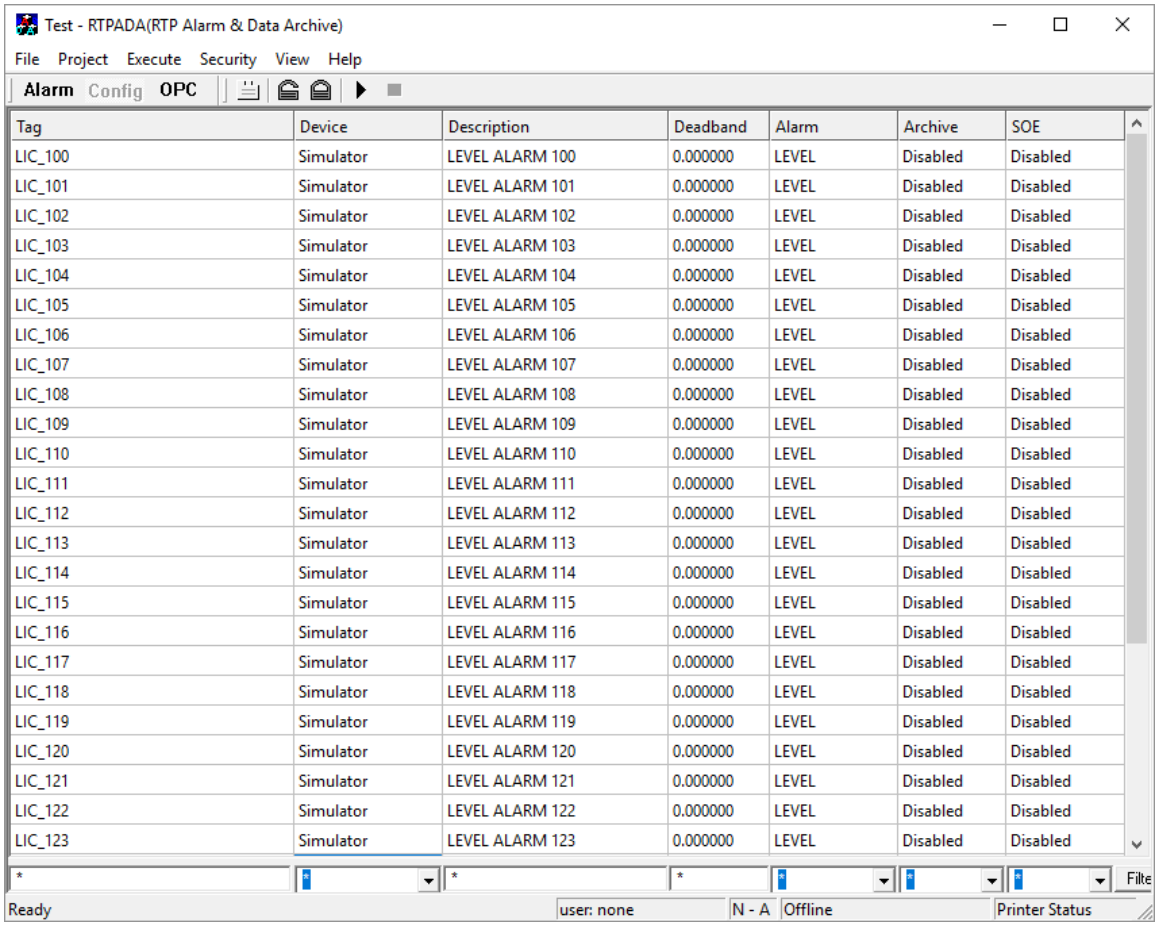
First we must create the RTP ADA file.

Importing Tag List Configuration into RTPADA

The tag table list .csv file created earlier simplifies the alarm point configuration.
This table is imported directly into the RTPADA (Alarm and Data Archive) system.

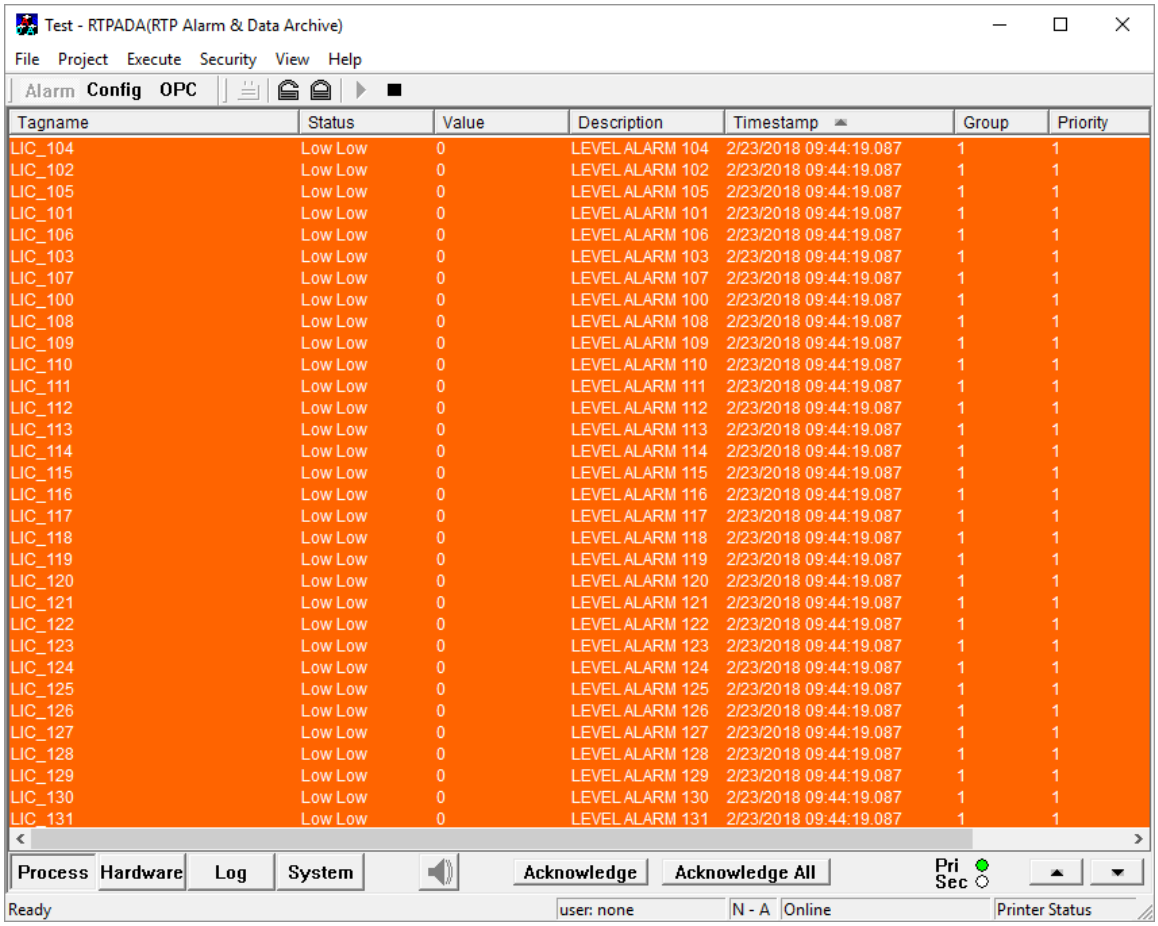
- Navigating to start button, programs, RTP NetSuite, select RTPADA
- Main menu, Click on File and select Import Configuration
- Select the Import_Table_1.csv file previously saved
- Select Device and Select Simulator for the first Tag.
- Right Click on Simulator, Select Copy, Left Click on the first unattached Device, Shift Left Click on the last Device in the group, Right Click and Select Paste.
- All of the Tags are now assigned to the Simulator.
- Select File Save Project as TEST.db

The alarm configuration table will appear as shown below.



Tag	Device	Description	Deadband	Alarm	Archive	SOE
LIC_100	Simulator	LEVEL ALARM 100	0.000000	LEVEL	Disabled	Disabled
LIC_101	Simulator	LEVEL ALARM 101	0.000000	LEVEL	Disabled	Disabled
LIC_102	Simulator	LEVEL ALARM 102	0.000000	LEVEL	Disabled	Disabled
LIC_103	Simulator	LEVEL ALARM 103	0.000000	LEVEL	Disabled	Disabled
LIC_104	Simulator	LEVEL ALARM 104	0.000000	LEVEL	Disabled	Disabled
LIC_105	Simulator	LEVEL ALARM 105	0.000000	LEVEL	Disabled	Disabled
LIC_106	Simulator	LEVEL ALARM 106	0.000000	LEVEL	Disabled	Disabled
LIC_107	Simulator	LEVEL ALARM 107	0.000000	LEVEL	Disabled	Disabled
LIC_108	Simulator	LEVEL ALARM 108	0.000000	LEVEL	Disabled	Disabled
LIC_109	Simulator	LEVEL ALARM 109	0.000000	LEVEL	Disabled	Disabled
LIC_110	Simulator	LEVEL ALARM 110	0.000000	LEVEL	Disabled	Disabled
LIC_111	Simulator	LEVEL ALARM 111	0.000000	LEVEL	Disabled	Disabled
LIC_112	Simulator	LEVEL ALARM 112	0.000000	LEVEL	Disabled	Disabled
LIC_113	Simulator	LEVEL ALARM 113	0.000000	LEVEL	Disabled	Disabled
LIC_114	Simulator	LEVEL ALARM 114	0.000000	LEVEL	Disabled	Disabled
LIC_115	Simulator	LEVEL ALARM 115	0.000000	LEVEL	Disabled	Disabled
LIC_116	Simulator	LEVEL ALARM 116	0.000000	LEVEL	Disabled	Disabled
LIC_117	Simulator	LEVEL ALARM 117	0.000000	LEVEL	Disabled	Disabled
LIC_118	Simulator	LEVEL ALARM 118	0.000000	LEVEL	Disabled	Disabled
LIC_119	Simulator	LEVEL ALARM 119	0.000000	LEVEL	Disabled	Disabled
LIC_120	Simulator	LEVEL ALARM 120	0.000000	LEVEL	Disabled	Disabled
LIC_121	Simulator	LEVEL ALARM 121	0.000000	LEVEL	Disabled	Disabled
LIC_122	Simulator	LEVEL ALARM 122	0.000000	LEVEL	Disabled	Disabled
LIC_123	Simulator	LEVEL ALARM 123	0.000000	LEVEL	Disabled	Disabled

- Click the play button  to start the alarm manager
- Click the Alarm tab to display active alarms



Add Objects to the RTP_ADA Module Form to provide signals for RTPADA Example.

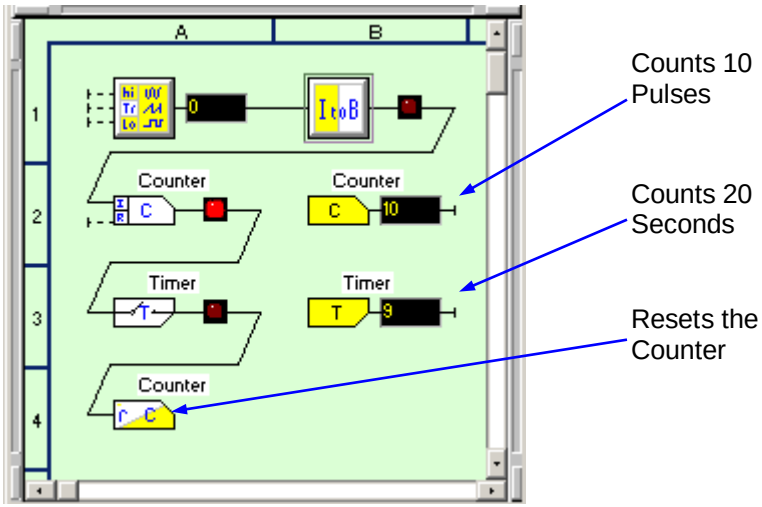
- Return to the Main Form by clicking on the  button in the Main Toolbar.
- Double-click on the **RTP_ADA** object in the Main Form.
- Click on the **Yes** button to create a new Module Form with the Tag **RTP_ADA**.
- Go to Module Form MForm1 and copy as is the float integers in cells A8 through A16. Paste the objects in the RTP_ADA form in cells B2 through B10.
- Go to Module Form MForm1 and copy as is the float integers in cells B1 through B7. Paste the objects in the RTP_ADA form in cells D2 through D8.
- Go to Module Form MForm1 and copy as is the float integers in cells B8 through B16. Paste the objects in the RTP_ADA form in cells F2 through F10.
- Go to Module Form MForm1 and copy as is the float integers in cells C1 through C7. Paste the objects in the RTP_ADA form in cells H2 through H8.

- Select a Float Constant object and place in cell A1. Change the tagname to 10 and initial value to 10.
- Select a Float Waveform Generator and place in cell A2. Change the period to 30. Connect the max input of the waveform generator to the integer constant in cell A1. Connect to the output of the waveform generator to each float variable object in column B.
- Select a Float Waveform Generator and place in cell C2. Select wave type of Triangle. Change the period to 30. Connect the max input of the waveform generator to the integer constant in cell A1. Connect to the output of the waveform generator to each float variable object in column D.
- Select a Float Waveform Generator and place in cell E2. Select wave type of Ramp Up. Change the period to 30. Connect the max input of the waveform generator to the integer constant in cell A1. Connect to the output of the waveform generator to each float variable object in column F.
- Select a Float Waveform Generator and place in cell G2. Select wave type of Ramp Down. Change the period to 30. Connect the max input of the waveform generator to the integer constant in cell A1. Connect to the output of the waveform generator to each float variable object in column H.
- In the NetArrays File menu, select **Save My_first.dbn** to save your project.

This completes the creation of the NetArrays Project Program.

Verify Module Form Counter

- Display the Counter Module Form by clicking on the button in the Main Toolbar and then double-clicking on the **Counter** object in the Main Form.
- Observe that the output of the Waveform Generator alternates between 0 and 100, and that the output of the Integer to Bool converter switches between False and True every second.
- Observe that the Integer Counter Accumulator increments from 0 to 10, and when it reaches 10 the output of the Bool Counter object goes True.
- Observe that when the input to the Bool Timer goes True, the Integer Timer Accumulator counts from 0 to 20 seconds. When the Integer Timer reaches 20 seconds, its output goes True momentarily and resets the Integer Counter Accumulator. When the Integer Counter Accumulator gets reset, the Bool Timer freezes with an output to of 20 until the input again goes True. At that time the Integer Timer Accumulator resets to 0 and again counts up to 20 seconds.



If the counter/timer circuit does not behave as expected, stop the Simulator and carefully recheck the configuration of the objects within the Counter Module Form.

Verify Module Form RTPView

- Display the RTP_View Module Form by clicking on the button in the Main Toolbar and then double-clicking on the **RTP_View** object in the Main Form.
- Verify that the Integer Waveform in Cell B2 is cycling between 0 and 3, stepping approximately every half of a second.
- The rest of this module will be further utilized in the RTPView Exercise.

Verify Module Form RTP_ADA

- Display the RTP_ADA Form by clicking on the button in the Main Toolbar and then double-clicking on the **RTP_ADA** object in the Main Form.
- Verify that all the Float Variables with tagnames beginning with LIC are changing.